Computer vision Seminar

수원대학교

Data Network Analysis 

데이터과학부

정인호

2023.02.09

Week5) GoogLeNet, ResNet

1. Week 4 과제 리뷰
2. GoogLeNet(2014)
3. ResNet(2015)
4. Week 5 과제 설명

수원대학교
Data Network Analysis

Week5) GoogLeNet, ResNet

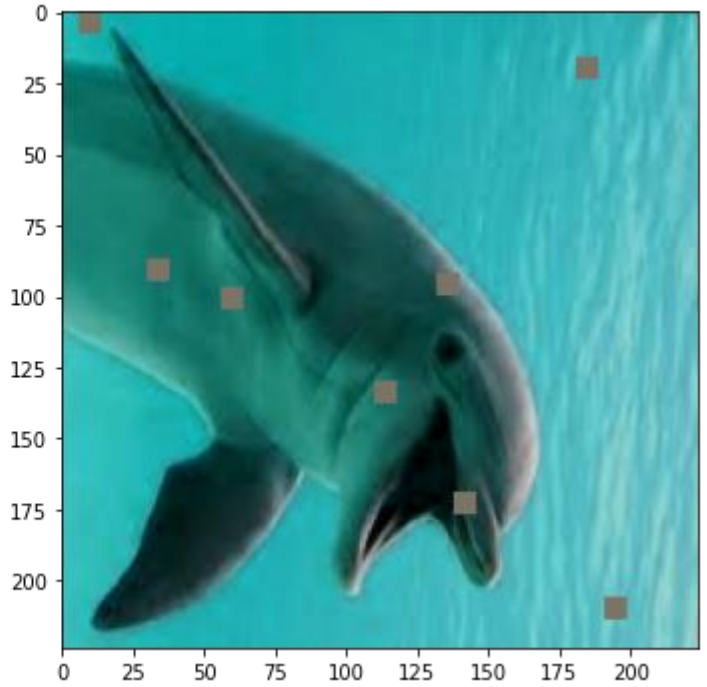
1. Week 4 과제 리뷰

1. Week 4 과제 리뷰

과제 피드백 – 가독성이 높은 코드

[Week 4 과제풀이]

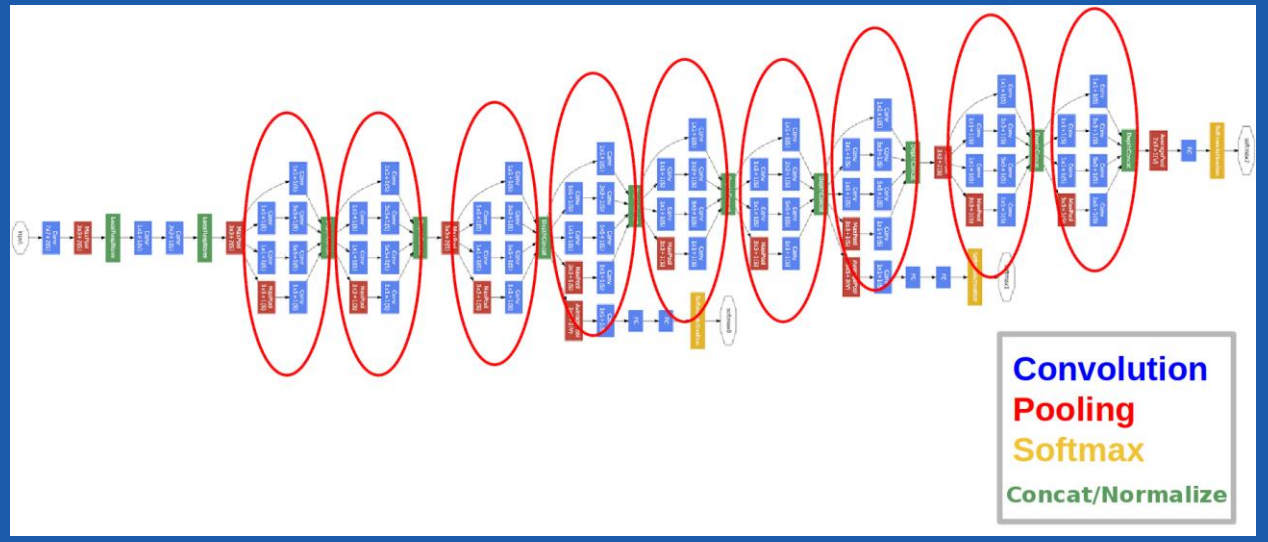
Raw Image



Transformed Image

Week5) Pytorch, GoogLeNet

2. GoogLeNet(2014)



수원대학교
Data Network Analysis

2. GoogLeNet(2014)

GoogLeNet 논문

논문 초록을 읽어보자.

구글넷 블로그 포스팅: https://inhovation97.tistory.com/45

2. GoogLeNet(2014)

< 2. 저자들의 실험 초점 >

저자들이 연구한 내내 신경썼던 방향성이 있을 겁니다.

2. related work & **3. Motivation and High Level Considerations** 부분의 핵심 내용을 설명합니다.

(2. related work의 1x1 Conv와 다양한 사이즈의 ConV의 내용이 대략적으로 나오지만 아키텍쳐 부분에서 설명합니다.)

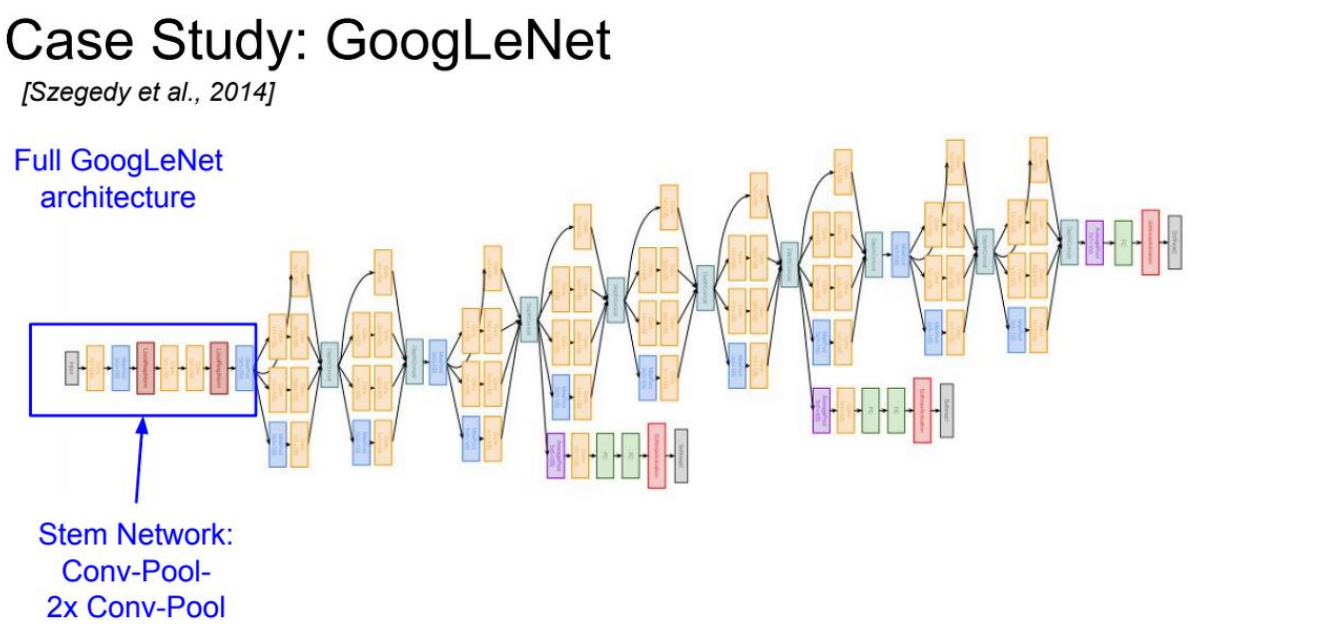
구글팀도 모델을 깊게 쌓기 위해서 노력을 하는데, depth와 width를 언급합니다. 아키텍쳐의 그림을 보면 아시겠지만, 여러 사이즈의 Conv를 쓰기 때문에 **width**라는 단어가 나옵니다.

모델을 깊게 쌓기위한 문제점

1. 모델을 깊게 쌓으면, 당연히 오버피팅의 문제가 존재합니다.
또한 모델이 너무 깊으면 모델의 일반화도 힘들어져서 학습셋이 적은 경우는 이미지의 디테일한 부분을 잡기가 더 더욱 힘들어집니다.

2. 네트웍의 사이즈가 깊어질수록 연산량은 훨씬 가파른 기울기로 증가합니다.
3x3의 Conv를 2개 쌓는다고 예를들면,
HxWxC가 3x3을 거쳐 HxWxC1로 나옵니다. 이걸 다시 Conv를 거치면 HxWxC2가 되는데, C1채널로 나온 피쳐맵을 또 다시 C2로 바꿔주니까 layer를 한개 더 쌓았을 뿐인데 계산량은 제곱으로 늘어나는거죠.
네트웍을 깊게 쌓되, 효율적으로 깊게 쌓아야 한다는겁니다.

2. GoogLeNet(2014)



처음 부분은 AlexNet의 구조를 그대로 가져갑니다.

vggnet과 큰 차이점은 vggnet은 처음에 큰 사이즈의 filter로 합성곱을 하는 것에 부정적이었습니다.

> vggnet의 configuration은 이제껏 우승했던 모델(AlexNet)과는 다르다고합니다.
> 첫 Conv layer를 비교적 큰 사이즈인 7x7, strid는 2를 쓴 AlexNet과는 달리 vggnet 저자들은 전체 모델에 stride는 1이며, 아주 작은 3x3필터를 쓴게 포인트라고 합니다.
> https://inhovation97.tistory.com/44

vggnet은 모델의 효율성을 위해 작은 필터만을 고집했지만, 구글넷 저자들은 관점이 달랐습니다.

input에 가까운 layer들은 굉장히 적은 범위에 correlated unit들이 분포하고 집중돼있다고 합니다.

이걸 제 생각대로 해석해보자면, 이미지 사이즈는 input일 때에 가장 사이즈가 큽니다.

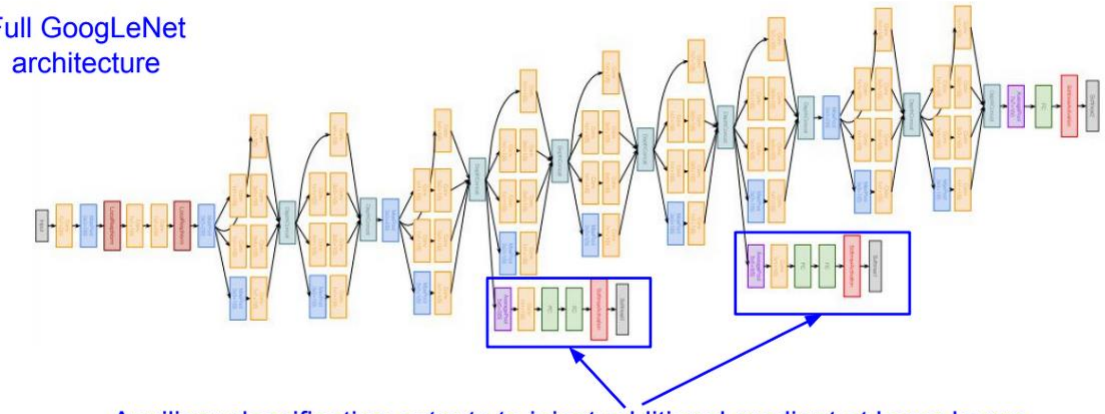
그러다보니, 필터들이 그것을 전부 훑으면, 낭비되는 값들이 당연히 많을 겁니다. 그래서 AlexNet처럼 일단은 resolution을 큰 filter로 크게 줄여버리고나서 본격적인 인셉션 모듈로 들어가자는 의도였던 것 같습니다.

2. GoogLeNet(2014)

Case Study: GoogLeNet

[Szegedy et al., 2014]

Full GoogLeNet architecture



중간 중간에 손잡이 모양의 분류기가 있습니다.

저자들은 **Auxiliary classifier**라고 하는데, 이것을 추가함으로써 backward 과정에서 깊은 네트워크의 기울기 소실 문제를 방지할 수 있습니다.

이 보조 분류기에도 똑같이 softmax 1000 class output을 설정하는데,

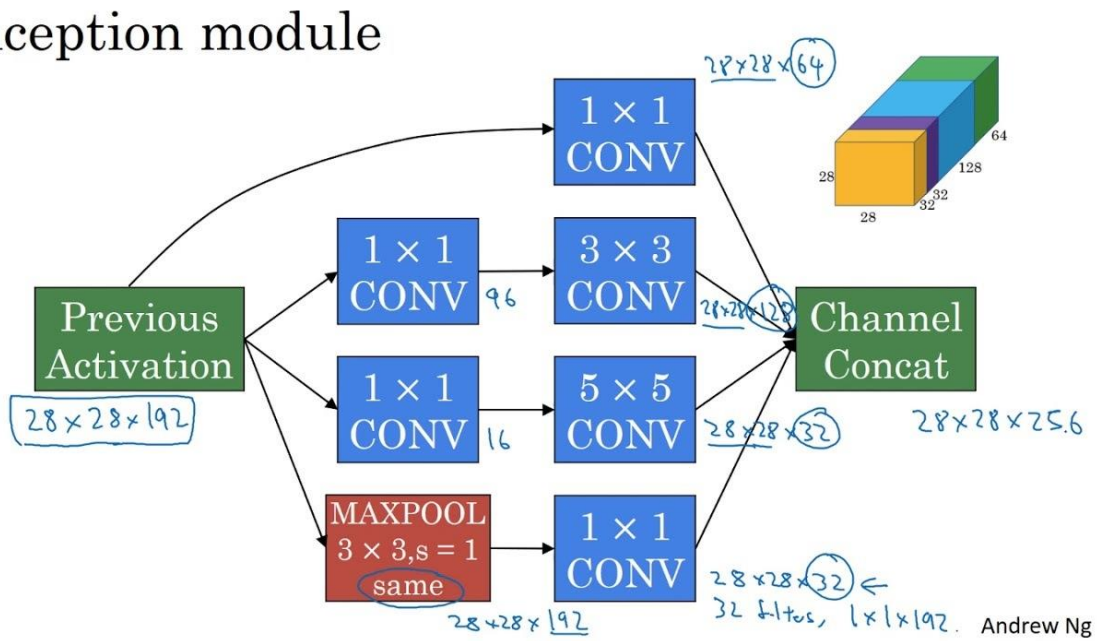
train시에는 Auxiliary classifier에서의 loss가 total loss에 더해지는데 0.3의 가중치를 곱해 영향력을 낮춰서 더해줍니다. 그렇기 때문에 저자들은 regularization의 효과도 기대해 볼 수 있다고 합니다. 약간의 앙상블(?) 효과도 기대해 볼 수 있는거죠.

추론 시에는 쓰지 않는다고합니다.

학습때만 이용합니다.

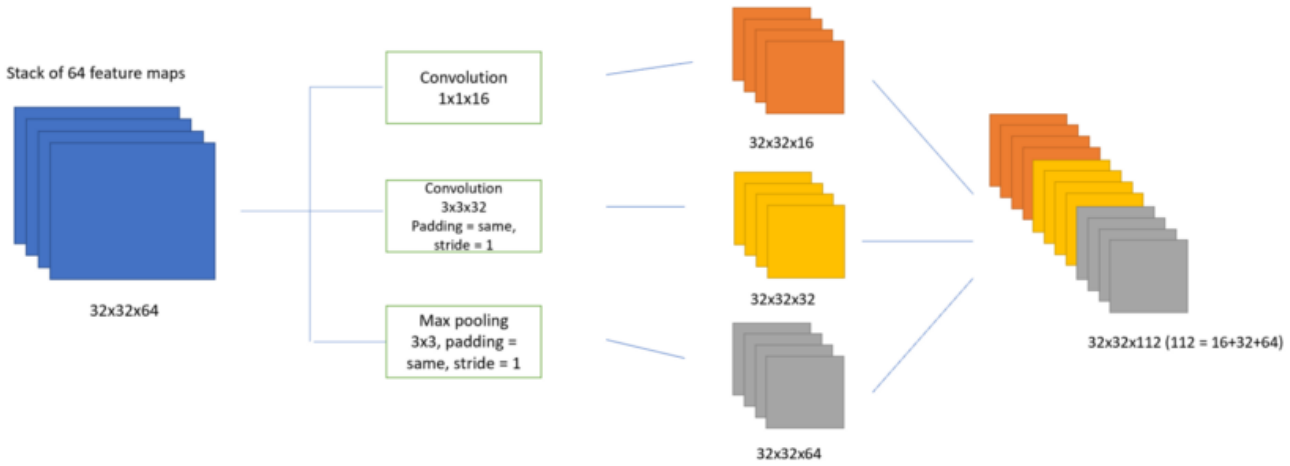
보조 분류기, 최종 분류기를 보면 **Global Average Pooling**이 쓰입니다.

2. GoogLeNet(2014)



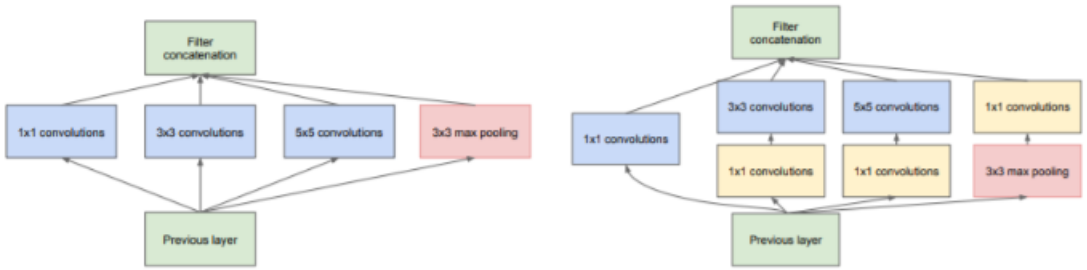
CNN width를 넓힌 GoogLeNet

2. GoogLeNet(2014)



https://valentinaalto.medium.com/understanding-the-inception-module-in-googlenet-2e1b7c406106

2. GoogLeNet(2014)



(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 2: Inception module

그렇게 인셉션 모듈을 만들고, naive version이 아닌 b를 선택하게 됩니다.

5x5 filter를 사용하면, 연산량이 너무 커지게 됩니다. 그래서 오른쪽과 같이 1x1 Conv로 channel reduction을 해줍니다.

1x1 filter가 vggnet에서도 그렇고 굉장히 중요한 역할을 한다는 것을 알 수 있습니다.

위 방식은 Cross Channel Pooling방식입니다.

maxpooling을 채널별 픽셀로 해줌으로써 같은 사이즈의 feature map을 원하는 사이즈로 down시킬 수 있는 기법입니다.

1x1 Conv는 학습할 수 있는 파라미터가 있습니다. 즉 down size도 좀 더 의미있게 할 수 있고, relu 함수도 거치기 때문에 비선형성도 한번 더 가해줄 수 있습니다. 여러 필터 사이즈로 나온 feature map들을 채널 수를 줄일 때 또 한 번 그것에 대해 고려해줄 수 있는 비선형성을 고려해줄 수 있으니 아주 좋은 방식인 거죠.

3x3 max pooling은 좀 더 다양한 resolution을 위해 추가해준 것이라고 합니다.

3x3 맥스 풀링을 할 때, 나와있지는 않지만 아마 사이즈 다운을 방지하기 위해 stride를 겹치게 할 겁니다.

그럴 경우 계속해서 특징이 강한 의미있는 픽셀의 값이 여러 번 겹치기 때문에 또 그런 부분의 장점이 있을 겁니다.

filter들의 output channel 같은 하이퍼 파라미터들은 layer depth마다 다르게 설정해줍니다.

(개인적으로는 이미 다른 발전된 모델은 보고나서 느낀 점은 이때의 구글넷이 너무 헤비해서 쓸모없는 정보도 많이 포함돼있지 않나 싶습니다. 이렇게 짜고도 비교적 간단한 vggnet과 크게 차이가 나지 않았고, backbone으로도 vggnet이 더 많이 쓰입니다.)

2. GoogLeNet(2014)

GoogLeNet 정리하기

Week5) ResNet(2015)

4. ResNet(2015)

ⓒ사진= 네이버 어린이백과[출처]

외국인에게 젓가락보다는 포크를...

두루미에게는 접시보다 호리병을...

수원대학교

Data Network Analysis

3. ResNet(2015)

[ResNet 논문]

논문 초록을 읽어보자.

ResNet 블로그 포스팅: https://inhovation97.tistory.com/46

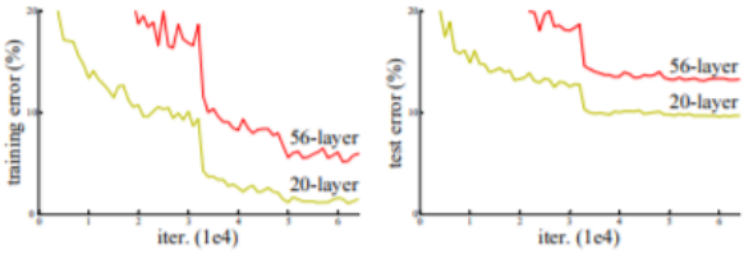
3. ResNet(2015)

"Is learning better network as easy as stacking more layers?"

"더 나은 네트워을 학습시키는 것이 레이어를 쌓는 것 만큼 쉬울까?"

(레이어를 증대하게 쌓은 네트워크의 학습은 쉬울 것인지에 대해서 고민)

이미 vanishing/exploding gradient의 문제가 있었지만, 이는 weight-initialization이나 중간중간의 normalization layer, SGD를 통해 깊은 네트워의 수렴을 이끌어 왔습니다.



하지만 위 fig를 보면, 56-layer 모델이 20-layer 모델보다 train/test error가 모두 높습니다.

56-layer 모델의 test error 뿐만 아니라 training error도 같이 증가했기 때문에 단순히 오버피팅의 문제도 아닙니다.

논문에서는 이를 deeper network이 수렴해갈 때 발생하는 **'degradation problem(of training accuracy)'** 이라고 언급합니다.

네트워이 더 깊어지면, 이 문제는 더 빨리 일어난다고 합니다.

(layer를 더 추가했더니 training error가 더 발생한 것)

이건 모든 시스템들이 유사하게 쉬운 수렴을 하는 것은 아니라는 뜻입니다.

따라서 저자들은 **'identity mapping'**이라는 솔루션을 생각합니다.

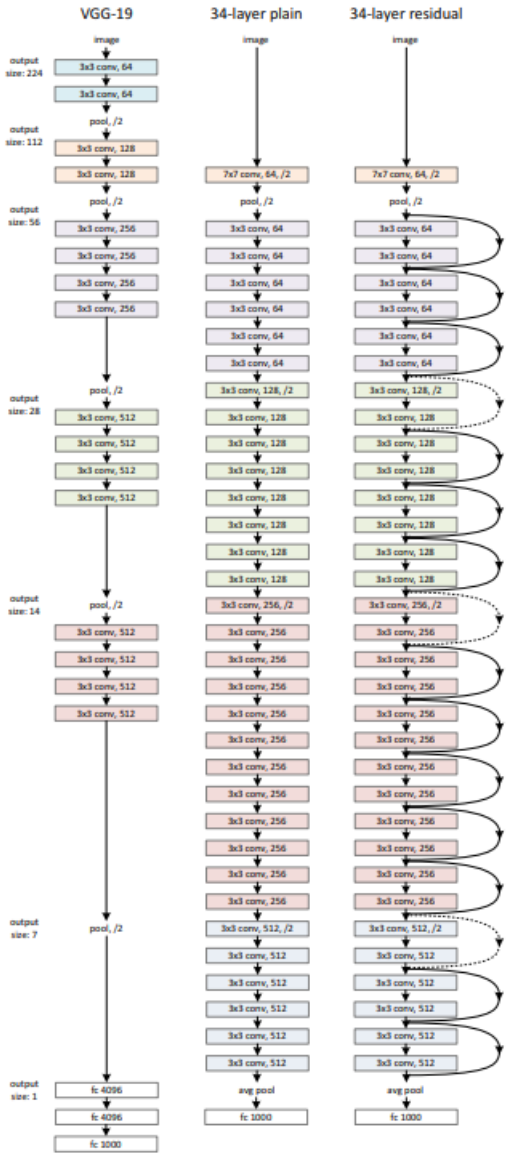
얕은 모델과 얕은 모델에서 layer들을 추가한 깊은 모델을 생각했을 때,

깊은 모델에서 얕은 모델의 똑같은 아키텍쳐 부분은 그대로 둔 채 추가된 layer는 단순히 identity mapping을 한다면,

적어도 깊은 모델의 training error는 얕은 모델보다 그 이하여야한다는 목적의 실험을 진행합니다.

하지만 실험 결과가 그닥 좋지 못했고, 다른 솔루션을 생각합니다.

~~(이 부분은 아래의 아이디어에 대한 identity mapping에 대한 가정으로 깔고 가는 부분인 것 같습니다. 셋업을 하는 느낌...?)~~



3. ResNet(2015)

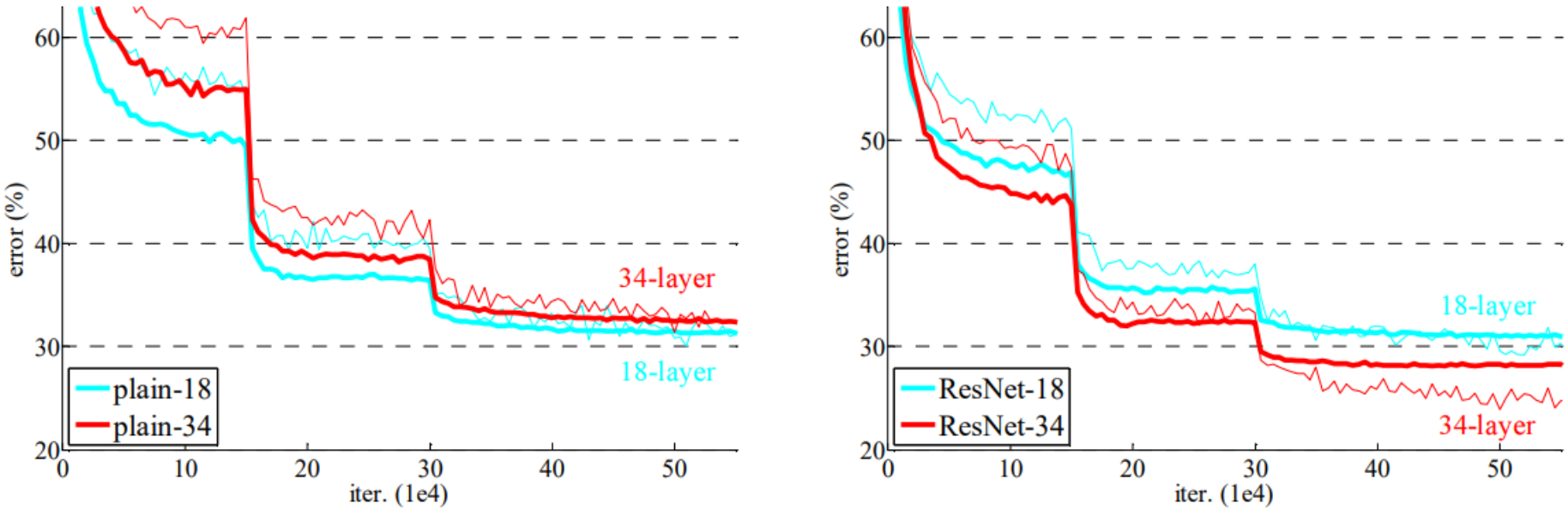


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

3. ResNet(2015)

그럼 Identity mapping을 알아보자.

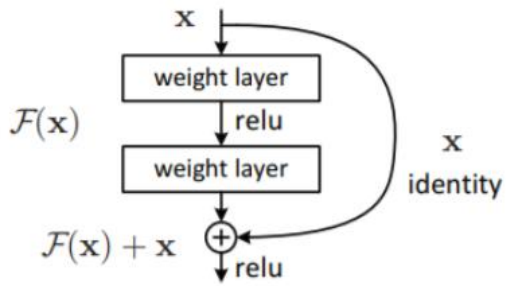


Figure 2. Residual learning: a building block.

identity mapping이 아니라 F(x)+x를 매핑하는 residual mapping을 생각해냅니다.

만약에 identity mapping이 최적으로 수렴한 상태라면, residual인 F(x)를 0으로 보내는 것이 더 쉬울 것이라는 것

입력값 x를 전달해주는 부분을 shortcut이라고 부르며, 이는 단순히 출력값 F(x)에 더해주는 것이기 때문에 추가적인

파라미터나 계산 복잡도를 높이지 않습니다. -> 이 부분은 아래 3에서 더 자세히 설명됨

(모든 레이어에서는 온전히 잔차 F(x)에 대한 학습만 하면 됩니다.

<순전파>

conv layer의 특성상 layer가 깊어질수록 이미지를 더욱 global하게 바라보고, high-level feature를 학습합니다. 이제

는 residual block을 쓰면, 각 레이어는 identity로 매핑받은 값으로 부터 이전 출력값과의 차이인 F(x)를 학습하는 것이

니 본인들의 임무가 더욱 명확해졌기 때문에 학습/수렴이 더욱 쉬워지는 것입니다.

<역전파>

shortcut 은 단순히 더하기 노드이기때문에 복잡도도 커지지 않으며, 역전파에서도 기울기를 그대로 전파할 수 있어서 g

radient vanishing의 문제도 해결됩니다.

)

<순전파>

<역전파>

3. ResNet(2015)

Residual Networks

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Table 2. Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts. Fig. 4 shows the training procedures.

table 2

residual net도 18-layer, 34-layer로 진행.

plain net을 베이스라인으로 똑같은 아키텍쳐로 3x3 필터마다 shortcut connection을 깔아줌. (fig. 3)

table 2와 fig. 4를 보면서, 주요한 상황 3가지를 관찰했습니다. - 결과

1. 34-layer residual net와 18-layer residual net이 plain net과는 달리 상황이 역전되었습니다.

심지어는 34-layer는 training error를 굉장히 많이 줄였고, validation data에 대해서 일반화를 상당히 잘 시킨 것을 알 수 있습니다. 이렇게 세팅한 것이 degradation문제에 대해 잘 짚었기때문에 모델을 깊게하면서도 accuracy gain을 할 수 있었던 것.

(degradation의 문제는 수렴에 대한 어려움의 문제가 맞았다.)

2. plain net과 비교해서 34-layer residual net은 trainin error와 더불어 top1 error를 3.5%까지 줄였습니다.

(fig. 4, table 2)

이는 deep모델에서 residual net의 굉장한 효과를 입증해줌.

(residual net은 모델의 깊이에만 관여하는 것이 아니라 같은 깊이에서도 plain net에 비해 성능이 더 좋다.)

3. residual net의 수렴 속도가 훨씬 더 빠르다.

fig. 4에서 plain net에 비해 residual net의 초기 수렴 기울기가 훨씬 가파른 것을 볼 수 있습니다.

간단한 fomula를 재정의해준 결과가 속도까지 관여했던 것.

(residual net의 수렴 속도가 더욱 빠르다.)

ResNet 정리.

Week5) GoogLeNet, ResNet

4. Week 5 과제설명

수원대학교
Data Network Analysis

4. Week 5 과제 설명

week 5
과제 설명

본인의 깃허브를 꾸며서 저한테 링크를 보내주세요.

발표자 정인호 깃허브